Joshua Kephart

GitHub username: jak243

Repository: https://github.com/jak243/CS1632\_d3.git

CS 1632 – DELIVERABLE 3: Performance Testing

Summary:

Honestly the most challenging thing that I encountered was getting flamegraph to work at all. I kept on running into a weird issue with the gemfile. Once I finally got it working, it looked a lot less helpful than I expected, but I am not sure whether my implementation is just really that bad (I don’t think it is) or if I just really don’t understand flamegraph.

I considered the edge cases and failure modes given directly in the description as well as an obvious case of a line being split up by “|” in an incorrect number of sections. I then used a generic rescue to cover any issues caused by a line or file being formatted incorrectly or a file not existing.

So it appears from the flame graph for long.txt that my hash function is taking up a lot of time, as well as every method which uses .each to search through my array of addresses. I have not made any changes based on these because it does not seem horribly slow to me, I am not sure I am reading the flamegraph correctly, and if hash IS the problem, there isn’t much I can do to speed that up.

Timing:

Running the program with long.txt (while also generating flamegraph) I found these three times:

27.0924018 seconds

28.0944968 seconds

29.0916433 seconds

The MEDIAN = 28.0944968 seconds

The MEAN = 28.0928473333 seconds

Flamegraph Screenshot: Here you can see what I was saying, this isn’t what I expected at all.

